(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 840 276 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.05.1998 Bulletin 1998/19

(51) Int. Cl.^{6}: **G09G 5/14**

(21) Application number: **97119122.6**

(22) Date of filing: **03.11.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **01.11.1996 US 30107 P**

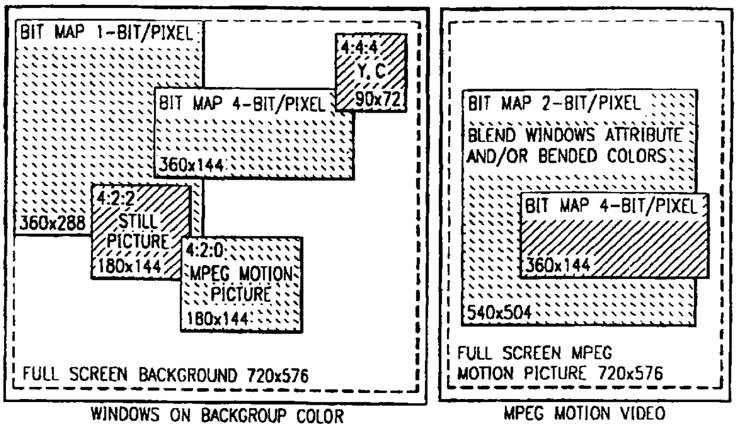
(71) Applicant:
TEXAS INSTRUMENTS INCORPORATED
Dallas, Texas 75243 (US)

(72) Inventors:
- **Chauvel, Gerard**
  300 Ch de la Suquette, 06600 Antibes (FR)
- **Benbassat, Gerard**
  06570 St-Paul-de-Vence (FR)
- **Chae, Brian**
  Plano, TX 75075 (US)

(74) Representative: **Holt, Michael**
Texas Instruments Limited,
Kempton Point,
68 Staines Road West
Sunbury-on-Thames, Middlesex TW16 7AX (GB)

(54) Window processing in an on screen display system

(57) A system is described that allows simultaneous display on a display screen of bit-map, graphic, still video picture, motion video picture or background. A frame memory containing the page to be displayed is located in an the SDRAM. A display controller reads the frame memory block by block and transfers the data to a Fifo. For each pixel, the OSD decoder reads the bits required to display the current pixel from the FIFO. The number of bits per pixel varies during the display depending upon the mode. The pixel selector and its controller select the bits of data from the FIFO to form the current pixel.


BIT MAP 1-BIT/PIXEL
360x288
BIT MAP 4-BIT/PIXEL
360x144
4:4:4
Y, C
90x72
4:2:2
STILL
PICTURE
180x144
4:2:0
MPEG MOTION
PICTURE
180x144
FULL SCREEN BACKGROUND 720x576
BIT MAP 2-BIT/PIXEL
BLEND WINDOWS ATTRIBUTE
AND/OR BENDED COLORS
BIT MAP 4-BIT/PIXEL
360x144
540x504
FULL SCREEN MPEG
MOTION PICTURE 720x576

WINDOWS ON BACKGROUP COLOR

MPEG MOTION VIDEO
AND BLEND WINDOW

FIG. 3

EP 0 840 276 A2

Printed by Xerox (UK) Business Services
2.16.1/3.4

Description

FIELD OF THE INVENTION

This invention relates to an on-screen display system with variable resolution.

BACKGROUND OF THE INVENTION

Current OSD systems may employ multiple central processing units (CPUs) in order to successfully deal with and handle high speed digital bit streams, such as those associated with a digital television set-top box. Each such CPU requires at least its own working memory space. These systems are expensive and may require expensive high speed memories.

The OSD co-processor of the present invention provides for OSD systems that overcome these and other shortcomings of existing OSD systems.

SUMMARY OF THE INVENTION

The present invention provides an OSD co-processor that allows for an OSD system having simultaneous display on a display screen of combinations of bit-map, graphic, still video picture, motion video picture or background. A frame memory containing a page to be displayed is preferably located in an external memory, which is preferably SDRAM. The OSD co-processor includes a display controller, a FIFO, an attribute memory, a window controller, a decoder, and line and pixel counters; the co-processor may also include an address calculator. A display controller reads the frame memory block by block and transfers the data to a FIFO. For each pixel, the decoder reads the bits required to display the current pixel from the FIFO; the number of bits per pixel may vary during the display depending upon the mode. A pixel selector and its controller select the bits of data from the FIFO to form the current pixel.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be further described, by way of example, with reference to the example with reference to the accompanying drawings in which:

Figure 1 depicts a high level architectural diagram of an audio/video decoding system employing an OSD co-processor of the present for to displaying an OSD picture;
Figure 2 depicts display modes of an OSD co-processor of the present invention and their associated memory requirements;
Figure 3 depicts two representative OSD pictures generated by an OSD co-processor of the present invention;
Figure 4 depicts how a CPU builds windows in a portion of its memory and the windows are used to build a frame in a frame memory which is displayed by an OSD co-processor of the present invention on a screen display;
Figure 5 depicts a high level architectural block diagram of an OSD co-processor of the present invention and selected interconnections;
Figure 6 depicts two windows having different color schemes that illustrate how an OSD co-processor of the present invention displays two overlapped windows;
Figure 7 depicts how a portion of the windows of Figure 6 are stored in a frame memory;
Figure 8 depicts in more detail selected portions of the blocks of Figure 5;
Figure 9 depicts a portion of a window controller of the present invention;
Figure 10 depicts portions of a memory cell and memory logic utilized in a window controller of the OSD co-processor of the present invention;
Figures 11-13 depict the generation, assembly and display of portions of windows depicted in Figures 6 and 7;
Figure 14a depicts representative window transition timings, such as those for Figures 6 and 7; and
Figure 14b depicts representative pixel selection and multiplexer timings for Figure 13.

DETAILED DESCRIPTION

The present invention provides an OSD co-processor for an OSD system. Figure 1 depicts one such OSD system.

Figure 1 shows the global flow for decoding and displaying an OSD picture. The PSO buffer contains the coded picture, for example: Teletext data to be displayed within an OSD picture. The ARM CPU decodes (1) the Teletext data and builds (2) the OSD picture using a Bitblt hardware accelerator. The OSD controller reads (3) the OSD buffer and generates the OSD video that is mixed with MPEG video (4).

Different modes may be displayed simultaneously on the screen: A bit-map window with 2 colors, a still video picture, a 256 colors graphic window, a decimated motion video picture and a Logo in true color. The OSD co-processor minimizes the memory required to display service information. The OSD uses a new hardware windowing technique to mix different modes of display such as: bit map with variable resolution, graphics, still video picture and decimated motion video.

Figure 2 depicts the various modes of display supported by the OSD coprocessor of the present invention.

More particularly, the OSD co-processor supports:

1) In a bit map the frame buffer contains the code of the color of each pixel to be displayed. The number of bits per pixel defines the number of the colors that can be displayed on screen. Two colors require 1-bit per pixel and 256 colors require 8-bits per pixel. This mode is used to display graphics and text. The code of the color addresses the Color Look Up Table (CLUT) that contains the three color components with 8-bits each.

2) Graphic plane, 4:4:4 mode, uses 8_bits per pixel for each color component R,G,B or Y,Cr,Cb. Each pixel requires 24-bits (16 million colors per pixel).

3) With the 4:2:2 mode the resolution of the chroma is divided horizontally by two. Each pixel has a luminance component Y and alternately a chrominance component Cr or Cb. This mode is used for video still or motion picture. Each pixel requires 16-bit.

With the 4:2:0 mode the resolution of the chroma is divided by two horizontally and vertically. Each pixel has a luminance component Y and a chrominance component Cr and Cb for 4 pixels. This mode is used by the MPEG video decoder to the size of the memory required to store a video motion picture. Each pixel requires 12-bit. The chroma interpolator generates the 4:2:2 output format.

Current techniques for OSD display:

Table 1 shows the number of bytes required to store a full screen picture 720 pixels x 576 lines in different display modes for an OSD system that does not allow mixing of several modes of display in real time on the same screen. When one part of the screen requires 256 colors, the full screen must be 256 colors even if a part of the screen requires only 2 colors.

Display mode	Number of bytes full screen
Bit map 1-bit per pixel	51840
Bit map 2-bit per pixel	103680
Bit map 4-bit per pixel	207360
Bit map 8-bit per pixel	414720
4:4:4 graphic plane	1244160
4:2:2 video mode	829440
4:2:0 video mode	622080

Table 1: Number of byte for full screen display

However, the OSD coprocessor of the present invention can display several modes described above simultaneously on the same screen. Each part of the screen with different display mode uses a windows fully programmable in size, in position and priority level. The windows can be overlapped without limitations. The characteristic of each window is defined by a set of attributes stored into a SRAM. Those attributes are used to calculate the memory address in real time during display and to control the hardware display system.

Windows on Background Color MPEG Motion Video and Blend Window

Window size		Number of bytes		Window size		Number of bytes
90 x 72	4:4:4	19 440		540 x 504	2-bit/pixel	68 040
360 x 288	1-bit/pixel	12 960		360 x 144	4-bit/pixel	25 920
360 x 144	4-bit/pixel	25920				
180 x 144	4:2:2	51840				
720 x 576	Background	0				
	Total	110160			Total	93 960
720 x 576	Full screen 4:4:4	1 244 160		720 x 576	Full screen 4-bit/pixel	207 360

Example of OSD Picture Memory Requirements for Pictures of Figure 3

Figure 3 shows typical OSD pictures. The left screen is composed of 5 OSD windows with different characteristics displayed on a full screen background color:

a bit-map window 360 pixels by 288 lines with two colors that requires 1-bit per pixel,
a bit-map window 360 pixels by 144 lines with 16 colors that requires 4-bit per pixel,
a graphic window 90 pixels by 72 lines with 16 million colors that requires 24-bit per pixel,
a still video picture window 180 pixels by 144 lines in 4:2:2 format using 24-bit per pixel,
an MPEG motion video window 180 pixels by 144 lines in 4:2:0 format. The window size and position is generated by the OSD system. The MPEG video decoder generates the content of the window by decimation of the full screen picture.

This OSD picture requires 110Kbytes memory. Without the OSD coprocessor the memory could be up to 1.24Meg bytes.

The right screen shows an other picture composed of 2 OSD windows displayed over full screen motion video:

a bit-map window 540 pixels by 504 lines with two colors that requires 1-bit per pixel. The background color is blended over motion video; and
a bit-map window 360 pixels by 144 lines with 16 colors that requires 4-bit per pixel.

The memory size is 93Kbytes with OSD coprocessor compare to 207Kbytes without OSD coprocessor.

Display flow:

In frame mode (figure 4) CPU and frame uses different memory area. The CPU build the windows separately into the CPU memory, each windows has its own display attributes i.e. display mode, resolution. The new display picture is created by the CPU by coping sequentially with each segment of the window 1 and 2 in such way that the OSD display controller reads the frame memory sequentially and display line by line from the left upper corner to the right lower corner. The display area that does not contain any OSD data, as background color or motion video are not described into the frame memory. For each transition of window, each line, he controller changes synchronously the attribute in order to display the window with the corresponding mode. The number of bits to display a pixel of the window 1 and 2 can be different.

The block diagram of the OSD coprocessor is shown figure 5:

Line counter: The pixel and line counter receives the pixel clock, generate X, Y that represent the pixel position on the screen and synchronization signals Hsyn and Vsync to control the screen display.

Windows Controller: contains the positions X and Y of each windows to be displayed on the screen. The controller compares X, Y position, and indicates to the display controller each window transition and window number.

Address Calculator: Not used in frame mode.

Display controller: For each new window transition the display controller read the new attribute from the attribute memory. It generates an address for the display memory and load the Fifo with a new block of data. It generate the attribute for the current window to the Decoder.

Decoder: For each pixel the decoder extracts from the Fifo the number of bits corresponding to the current pixel to be displayed. It transform data bit in pixel.

Windows attributes:

Display modes: empty window for decimated video. Bitmap, YCrCb 4:4:4 graphics component, YCrCb 4:2:2 CCIR 601 component and background color.

Supports blending of bitmap, YCrCb 4:4:4, or YCrCb 4:2:2 with motion video and with an empty window

Supports window mode and color mode blending

Provides a programmable 256 entries Color Look Up table

Outputs motion video for mixture with OSD in a programmable 422 or 444 digital component format

Provides motion video or mixture with OSD to the on-chip NTSC/PAL encoder

Each hardware window has the following attributes:

window position: any even pixel horizontal position on screen; windows with decimated video have to start from an even numbered video line also

window size: from 2 to 720 pixel wide (even values only) and 1 to 576 lines

window base address

data format: bitmap, YCrCb 4:4:4, YCrCb 4:2:2, and empty

bitmap resolution: 1, 2, 4, and 8 bits per pixel

full or half resolution for bitmap and YCrCb 4:4:4 windows

bitmap color palette base address

blend enable flag

4 or 16 levels of blending

transparency enable flag for YCrCb 4:4:4 and YCrCb 4:2:2

output channel control

Example of OSD display with 2 windows:

The figure 6 shows an example of display of two overlapped windows. The window 2 on back is a bit map with 16 colors per pixel. Each pixel require 4-bits into the frame memory to define the code of the color. The window 1 on top is a bit map with 2 colors per pixel. Each pixel require 1-bits into the frame memory to define the code of the color. The position and dimension of the windows 1 is given by the attributes X_10, X_11, Y_10, Y_11. Horizontally the number of pixel is: X_11 - X_10. Vertically the number of lines is: Y_11 - Y_10: Same for window 2 with X_20, X_21, Y_20 and Y_21.

The display controller access sequentially the display memory (figure 7) from the first word containing the pixel X_20, X_21 to the last word containing the pixel X_11, Y_11. Details of the line 20 is shown figure 7. The line begin with pixels of the window 2, window 1 start in X-10, Pa is the last pixel of window 2 with 4-bit per pixel, Pb is the first pixel of window 1 and use 1-bit per pixel. Window 1 end on Pc and window 1 restart in Pd until pixel position X_21.

The same word contain pixels of window 1 and 2. During display the window controller detect the transition between Pa and Pb and control the data processing algorithm after Fifo access.

Pixel selector and multiplex:

The pixel selector and multiplex is the input of the decoder block shown figure 5. It receive the outputs of the Fifo and the windows attributes and control signals from the display controller. The basic function is to transform data stored into the frame memory in pixel. In case of bit map display (figure 8) the output, Pixel_Mux[7:0] is the address of the CLUT (Color Look Up Table). For graphic or still video picture the output is one color component. Pixel_Mux[7:0] output is produced by bits coming from the Fifo and bits from attribute memory.

Pixel selector:

Receive 32-bit data from the Fifo outputs F[31:0]. The first stage select one of the five bytes F[31:24], F23:16], F[15:8], F[7:0] and F[6:0] delayed by one clock sample. The second stage is a half barrel shifter that allows to shift right a 15-bit input data by 0 to 7 position. The output of the barrel shifter position the LSB of the code of the pixel to be displayed in Bs[0]. The pixel counter provide the control signals for the multiplexor and barrel shifter. Table 2a shows the effect of the control Mux_S[1:0] on the bytes selection and 2b the output of the barrel shifter in function of Bs_S[2:0]. The table 2c combine the table 2a and 2b and shows the bits of the Fifo selected at the output of the barrel shifter in function of the 5_bit pixel counter.

Multiplexor:

Receive data Bs[7:0] from the barrel shifter and Base[7:0] from the attribute register. It is controlled by, 4 control bits coming from the attribute register, Cursor and default signals from display controller as shown table 2d.

Mux 5:1 x 8

Control Mux_S[1:0]	Byte Mux Outputs Bm[14:8]	Bm[7:0]
0 0	F-1[6:0]	F[31:24]
0 1	F[30:24]	F[23:16]
1 0	F[22:16]	F[15:8]
1 1	F[14:8]	F[7:0]

Barrel Shifter 15 to 8

Control Bs_S[2:0]	Bs[7:0] Outputs
0 0 0	Bm[7:0]
0 0 1	Bm[8:1]
0 1 0	Bm[9:2]
0 1 1	Bm[10:3]
1 0 0	Bm[11:4]
1 0 1	Bm[12:5]
1 1 0	Bm[13:6]
1 1 1	Bm[14:7]

Pixel Multipler outputs

Default	Cursor	Bit per Pixel Bp[3:0]				Mux output Pm[7:0]
		8	4	2	1	
1	0	X	X	X	X	Def[7:0]
X	1	X	X	X	X	Cur[7:0]
0	0	0	0	0	0	Base[7:0]
0	0	0	0	0	1	Base[7:1] & Bs[0]
0	0	0	0	1	1	Base[7:2] & Bs[1:0]
0	0	0	1	1	1	Base[7:4] & Bs[3:0]
0	0	1	1	1	1	Bs[7:0]

Table 2a, 2b and 2d: Pixel selector and multiplex control

Mux_S[1:0]	1 1	1 0	0 1	0 0
Bs_S[2:0]	Bs[7:0] =	Bs[7:0] =	Bs[7:0] =	Bs[7:0] =
1 1 1	F-1[6:0] & F[31]	F[30:23]	F[22:15]	F[14:7]
1 1 0	F-1[5:0] & F[31:30]	F[29:22]	F[21:14]	F[13:6]
1 0 1	F-1[4:0] & F[31:29]	F[28:21]	F[20:13]	F[12:5]
1 0 0	F-1[3:0] & F[31:28]	F[27:20]	F[19:12]	F[11:4]
0 1 1	F-1[2:0] & F[31:27]	F[26:19]	F[18:11]	F[10:3]
0 1 0	F-1[1:0] & F[31:26]	F[25:18]	F[17:10]	F[9:2]
0 0 1	F-1[0] & F[31:25]	F[24:17]	F[16:9]	F[8:1]
0 0 0	F[31:24]	F[23:16]	F[15:8]	F[7:0]

Table 2c: Pixel selector control tables.

Default: When active Pm[7:0] is equal to the 8-bit default color provided by the general control register of the display controller. No data is read from the Fifo.

Cursor: When active Pm[7:0] is equal to the 8-bit cursor color provided by the general control register of the display controller. No data is read from the Fifo.

Bp[3:0]=0000: The current window is empty and contain 0_bit per pixel or color component. Pm[7:0] is equal to Base[7:0] stored into the attribute register. In bit-map mode the Base[7:0] select one of the 256 colors of the CLUT as a background color.

Bp[3:0]=0001: The current window contain 1_bit per pixel or color component. Pm[7:0] is equal to Base[7:1] concatenated with Bs[0] from the barrel shifter. In bit-map mode the Base[7:1] is the base address of a set of 2_colors of the 256 colors CLUT.

Bp[3:0]=0011: The current window contain 2_bit per pixel or color component. Pm[7:0] is equal to Base[7:2] concatenated with Bs[1:0] from the barrel shifter. In bit-map mode the Base[7:2] is the base address of a set of 4_colors of the 256 colors CLUT.

Bp[3:0]=0111: The current window contain 4_bit per pixel or color component. Pm[7:0] is equal to Base[7:4] concatenated with Bs[3:0] from the barrel shifter. In bit-map mode the Base[7:4] is the base address of a set of 16_colors of the 256 colors CLUT.

Bp[3:0]=1111: The current window contain 8_bit per pixel or color component. Pm[7:0] is equal to Bs[7:0] from the barrel shifter. In bit-map mode the 256 colors CLUT are used.

Pixel counter:

Provide the control for the pixel selector Mux_S[1:0] concatenated with Bs_S[2:0]. Each beginning of frame the pixel counter is reset. It is decremented by 0, 1, 2, 4 or 8 depending of the current window attribute. Address clock signal is generated when the counter cross zero.

Memory address generator:

It generate the read address for the Fifo. Each address clock signal generated by the pixel counter a new 32-bit word F[31:0] is sent to the pixel selector.

Attribute memory and register:

The attribute memory contain the attributes of the windows to be displayed during the current frame. The attributes that control the pixel selector and multiplex are:

Display mode: bit-map, graphic, still video or empty.

Number of bit per pixel or color component: 0, 1, 2, 4 or 8 bits.

The bit-map CLUT base address.

The attribute register contain the attribute of the current window. The attribute clock transfers the content of attribute memory to the register when the window change.

Window controller:

The window controller is composed of a content addressable memory CAM, flip flop and priority encoder. The CAM contain the attributes of position and size of the windows to be displayed on the current frame. The figure 9 shows a 32 words CAM. Each window require 4 words that indicate horizontally the coordinates of the first and last pixel and vertically the coordinates of the first and last line. A 32_words CAM support 8 windows.

The CAM compare the value of the pixel and line counters respectively X and Y. When Y counter match a value Yn0, a new window start vertically, the corresponding RS flip-flop is set Wn_Y=1. When Y counter match a value Yn1, the corresponding window end vertically, RS flip-flop Wn_Y is reset. The process is the same horizontally. Wn_Y and Wn_X are combined to indicate that the X, Y counter is into an active window. The indices "n" indicate the window number. Several windows can be overlapped, the priority encoder indicate the display order on the screen. The indices "n=0" correspond to the cursor that must be always on top. Outputs of priority encoder are hit and OSD_W, Hit generates the attribute clock and OSD_W the memory attribute address corresponding to the active window.

The details of the CAM cell is shown figure 10. The cell contain 2 parts, a 6 Transistors RAM cell and a 4 transistors comparator. The CPU load the content of the CAM with the attributes of position of the windows, memory mode is selected. During the display the associative mode is selected. The line counter Y and pixel counter X are compared to the content of all Yn and Xn of the CAM. When X or Y is equal to one or several values the corresponding Match lines are active to set or reset a flip-flop.

Example of pixel generation:

The figures 11, 12 and 13 correspond to the generation of the display and frame memory represented figures 6 and 7.

Figure 11, the pixel Px of the window 1 is selected into the 32_bit word "m". The pixel counter is decrement by one each access. Pixel selector put Px at the output Bs[0]. The current window attribute select 1_bit per pixel. The pixel multiplex selects Base[7:1] and concatenate with Px to form the output Pixel_Mux[7:0]. In bit-map mode Pixel_Mux[7:0] select 2 colors of the CLUT. Base[7:1] is the base address of the CLUT.

Figure 12 shows the generation of a pixel Px for the window 2 that use 4_bits per pixel in bit-map mode. Pixel selector put Px at the output Bs[3:0]. The pixel multiplex selects Base[7:4] and concatenate with Bs[3:0] to form the output Pixel_Mux[7:0] that select one of the 16 colors of the CLUT. Base[7:4] is the base address of the CLUT. 16 CLUT of 16 colors can be selected with the window attributes.

Figure 13 shows a specific case of the window 2 where Pn+1 has its 2 msb bits into the word m+1 and 2 lsb into m+2. Word m+2 is currently selected at the output of the fifo and the 7 lsb bits of word m+1 are maintained into the 7_bit register F-1[6:0]. Pixel selector take F-1[1:0] concatenated with F[31:30] to generate Bs[3:0]. The corresponding timing is shown figure 14b. At Pn+1 pixel counter equal 30. The barrel shifter output generate F-1[5:0] concatenated with F[31:30]. The current fifo address is word m+2 and the Pixel_Mux output is Base[7:4] & F-1[1:0] & F[31:30]. Pixel Pn+2 the pixel counter is decrement by 4, the pixel selector select the next 4_bits of word m+2.

Window transition:

Figure 6 the window 1 is on top of window 2. The CAM of the window controller is loaded with window 1 attributes position on top of window 2. The window displayed on top has the lower indices. Pixel Pa is last pixel for this portion of window 2 into the frame memory, the next bits contain the code of color of window 1, the first bit is Pb. During the same line window 1 end in Pc and window 2 restart in Pd. Each of this window transitions are detected by the window controller that generate the attribute clock to change the current display attribute stored in a register (see timing figure 14a). For Pa-1 and Pa the Pixel counter is decrement by 4 each pixel and after attribute clock Bp[3:0]=1 (figure 8) decrement the counter by 1 each pixel.

Graphic and Still video modes:

In bit-map mode the frame memory contain the code of the color, 1,2,4 or 8 bits per color depending of the attributes. The code of the color concatenated with the CLUT color base attribute select a color from the CLUT. The CLUT contain 3 color components, one component for Luma and two components for Chroma.

When the current window is graphic, the CLUT is not used. Each pixel there is 3 access identical to the bit-map mode to extract the 3 colors components from the frame memory. There is 3 access during a pixel.

In still video picture the CLUT is not used. Still video reduce Chroma bandwidth. Each pixel there is 2 access identical to the bit-map mode to extract one Luma components and chroma component Cr or Cb alternately each pixel from the frame memory.

When the picture is empty the mode is bit-map with 0_bit per pixel. This mode is used to generate a window with a background color or a decimated motion video picture coming from the MPEG video decoder.

The pixel selector allows to mix by hardware different mode of display. The objective of this disclosure is to protect the pixel selector and multiplex hardware, its control, the window controller, window attribute generation and the flow of data from the frame memory to the Pixel_Mux output in the different display modes for different resolution.

The frame memory contain a description of the current displayed picture. The description is done sequentially starting from the first upper left corner of a window "n" to the last lower right corner of a window "m". Only the active windows need to be described, not the full screen, background color or motion video.

The description is done pixel per pixel, the number of bit to describe one pixel can vary for each window.

Each transition of window is indicated by the attributes of position stored into the CAM. The window controller select the attribute of the corresponding window in real time during display.

The pixel selector select the number of bits corresponding to the current pixel to be displayed.

The complementary bits to form the 8_bit pixel output are obtained by concatenation of selected data bits from the frame and window attribute from the attribute memory.

When the current window is bit-map Pixel_Mux output select a color of the CLUT. The CLUT contain the 3 colors component.

When the current window is graphic, the CLUT is not used. Each pixel there is 3 access identical to the bit-map mode to extract the 3 colors components from the frame memory.

In still video picture the CLUT is not used. Each pixel there is 2 access identical to the bit-map mode to extract the 2 colors components from the frame memory. Luma is extracted each pixel and chroma component Cr and Cb alternately each pixel.

When the picture is empty the mode is bit-map with 0_bit per pixel. This mode is used to generate a window with a background color or a decimated motion video picture coming from the MPEG video decoder.

Claims

1. An on-screen display co-processor, comprising:

a FIFO;
a decoder coupled to said FIFO;
a display controller coupled to said FIFO and decoder;
a window controller coupled to said display controller;
line and pixel counters coupled to said windows controller; and
an attribute memory coupled to said display controller.

2. A window controller, comprising:

a content addressable memory for storing data corresponding to window position and size and for comparing pixel and line counter data with such data,
at least one flip-flop coupled to said memory for indicating whether a window is active or not active, and
a priority encoder coupled to said flip-flop for selecting the display order of any active windows.

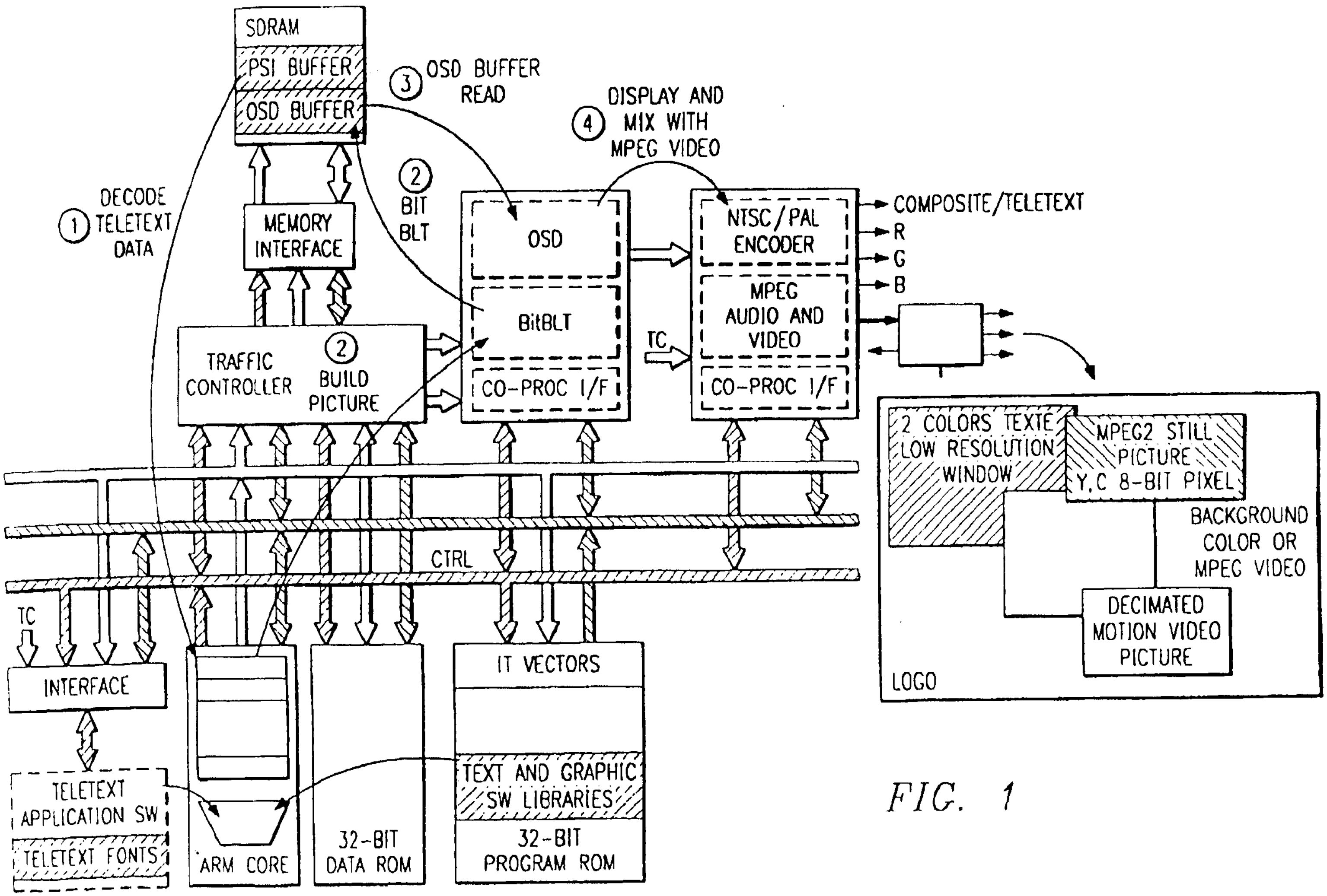

SDRAM
PSI BUFFER
OSD BUFFER
③ OSD BUFFER READ
④ DISPLAY AND MIX WITH MPEG VIDEO
① DECODE TELETEXT DATA
MEMORY INTERFACE
② BIT BLT
OSD
BitBLT
CO-PROC I/F
NTSC/PAL ENCODER
MPEG AUDIO AND VIDEO
CO-PROC I/F
COMPOSITE/TELETEXT
R
G
B
TC
TRAFFIC CONTROLLER
② BUILD PICTURE
CTRL
TC
INTERFACE
TELETEXT APPLICATION SW
TELETEXT FONTS
ARM CORE
32-BIT DATA ROM
IT VECTORS
TEXT AND GRAPHIC SW LIBRARIES
32-BIT PROGRAM ROM
2 COLORS TEXTE LOW RESOLUTION WINDOW
MPEG2 STILL PICTURE Y,C 8-BIT PIXEL
BACKGROUND COLOR OR MPEG VIDEO
DECIMATED MOTION VIDEO PICTURE
LOGO

FIG. 1

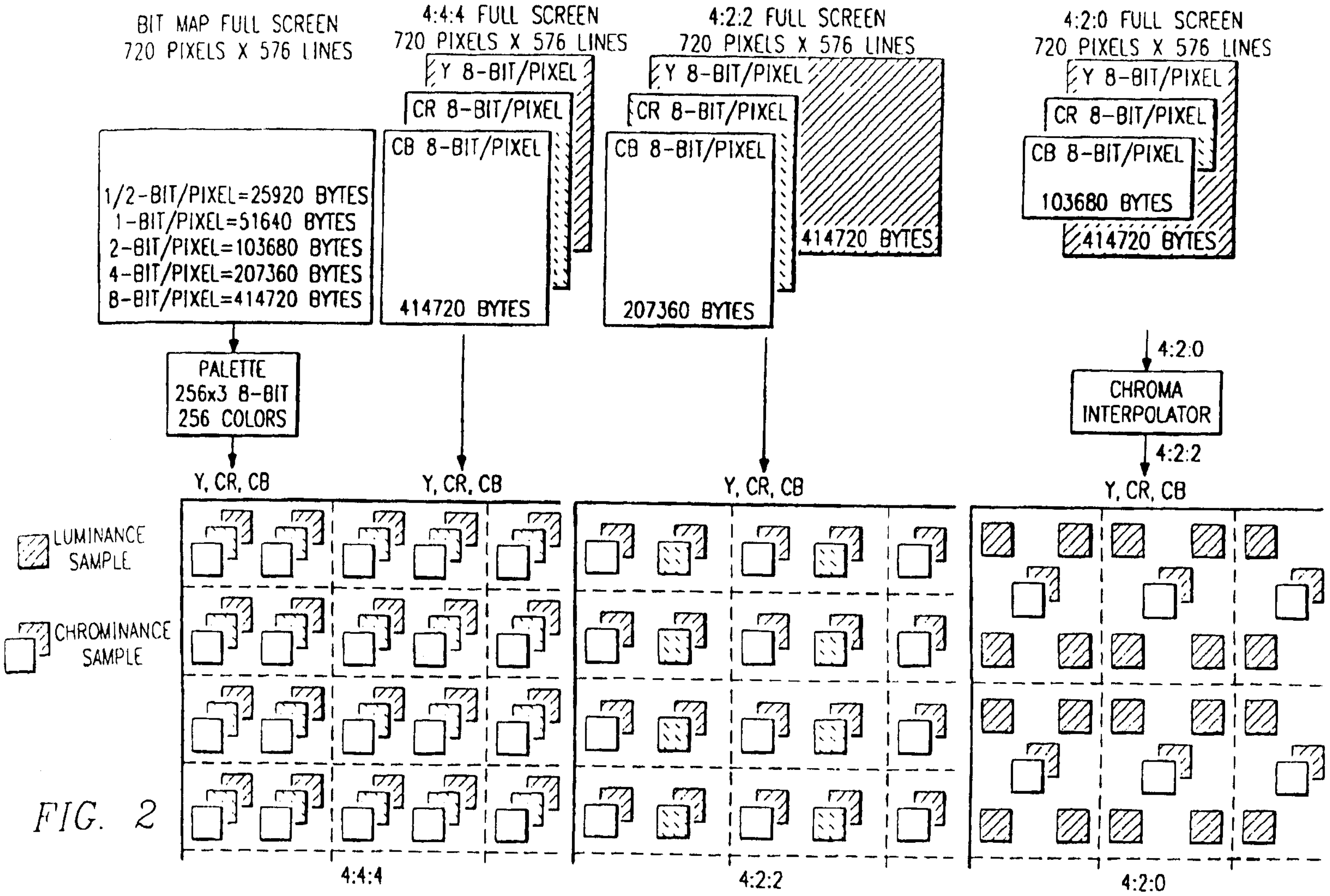

BIT MAP FULL SCREEN
720 PIXELS X 576 LINES
1/2-BIT/PIXEL=25920 BYTES
1-BIT/PIXEL=51640 BYTES
2-BIT/PIXEL=103680 BYTES
4-BIT/PIXEL=207360 BYTES
8-BIT/PIXEL=414720 BYTES
PALETTE
256x3 8-BIT
256 COLORS
4:4:4 FULL SCREEN
720 PIXELS X 576 LINES
Y 8-BIT/PIXEL
CR 8-BIT/PIXEL
CB 8-BIT/PIXEL
414720 BYTES
4:2:2 FULL SCREEN
720 PIXELS X 576 LINES
Y 8-BIT/PIXEL
CR 8-BIT/PIXEL
CB 8-BIT/PIXEL
207360 BYTES
414720 BYTES
4:2:0 FULL SCREEN
720 PIXELS X 576 LINES
Y 8-BIT/PIXEL
CR 8-BIT/PIXEL
CB 8-BIT/PIXEL
103680 BYTES
414720 BYTES
4:2:0
CHROMA
INTERPOLATOR
4:2:2
Y, CR, CB
Y, CR, CB
Y, CR, CB
Y, CR, CB
LUMINANCE
SAMPLE
CHROMINANCE
SAMPLE
4:4:4
4:2:2
4:2:0

FIG. 2

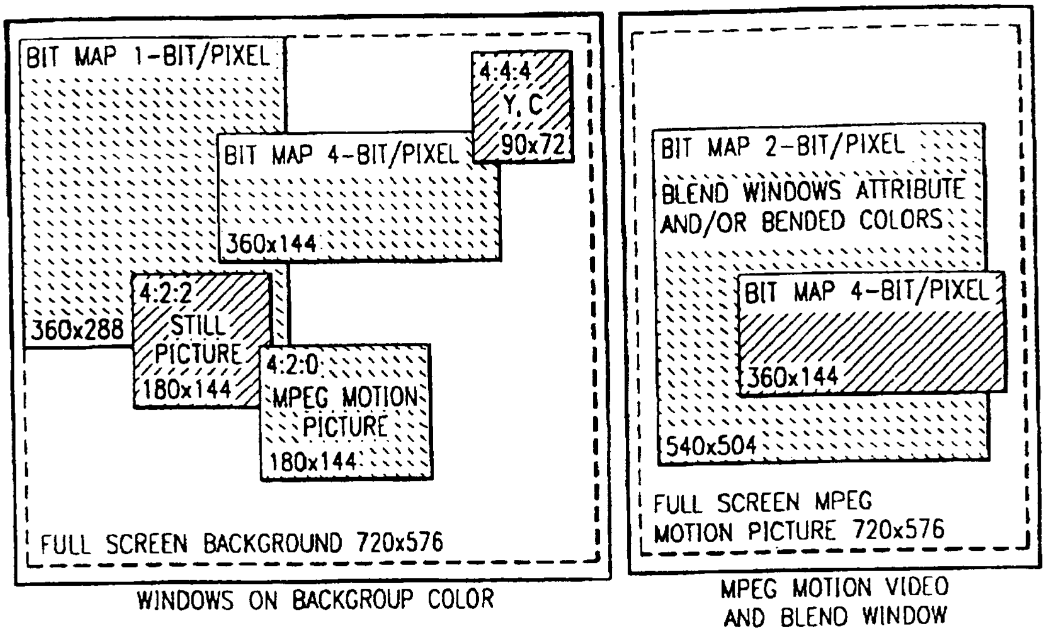

BIT MAP 1-BIT/PIXEL
360x288
BIT MAP 4-BIT/PIXEL
360x144
4:4:4
Y, C
90x72
4:2:2
STILL
PICTURE
180x144
4:2:0
MPEG MOTION
PICTURE
180x144
FULL SCREEN BACKGROUND 720x576
WINDOWS ON BACKGROUP COLOR
BIT MAP 2-BIT/PIXEL
BLEND WINDOWS ATTRIBUTE
AND/OR BENDED COLORS
BIT MAP 4-BIT/PIXEL
360x144
540x504
FULL SCREEN MPEG
MOTION PICTURE 720x576
MPEG MOTION VIDEO
AND BLEND WINDOW

FIG. 3

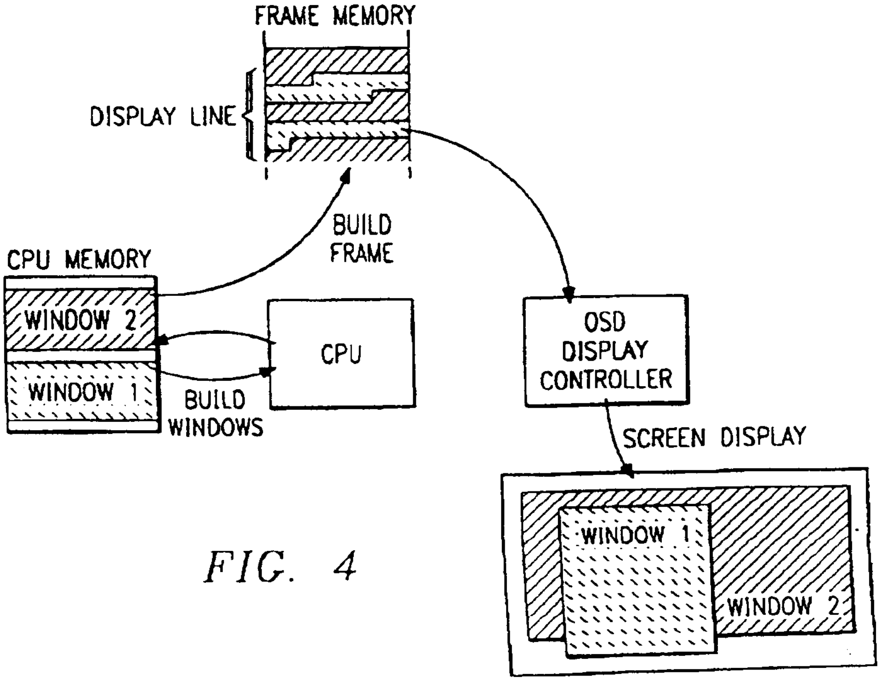

FRAME MEMORY
DISPLAY LINE
BUILD
FRAME
CPU MEMORY
WINDOW 2
WINDOW 1
CPU
BUILD
WINDOWS
OSD
DISPLAY
CONTROLLER
SCREEN DISPLAY
WINDOW 1
WINDOW 2

FIG. 4

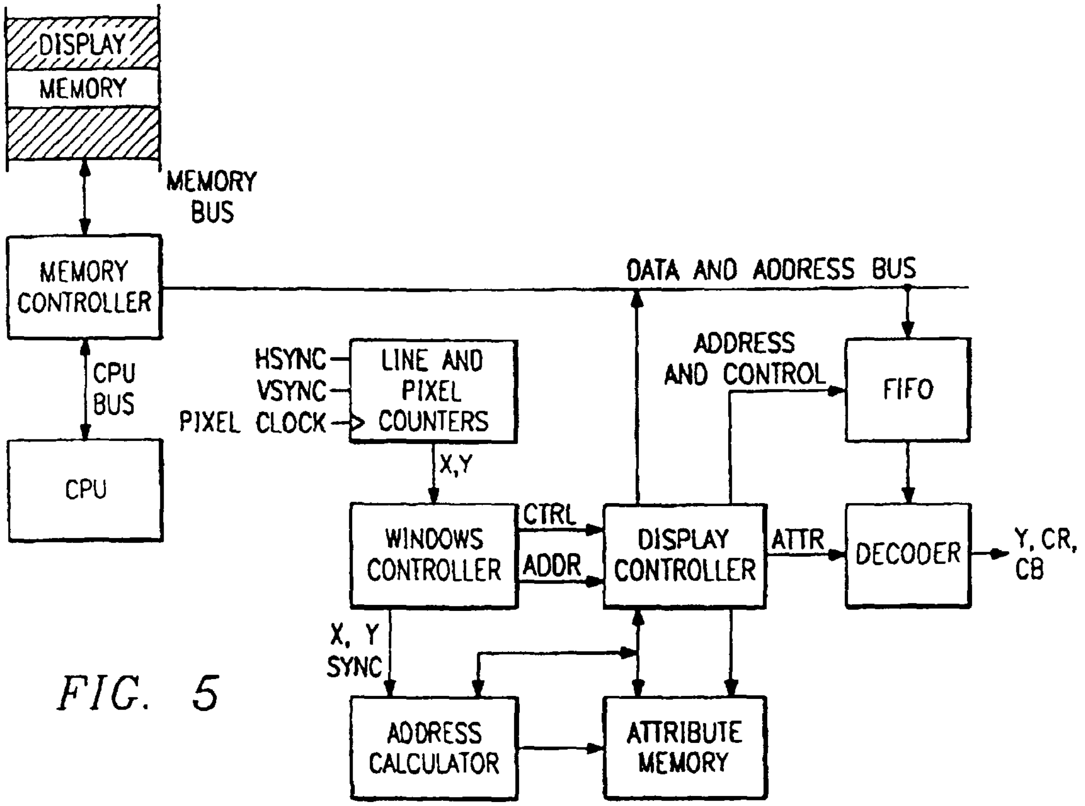

DISPLAY
MEMORY
MEMORY BUS
MEMORY CONTROLLER
CPU BUS
CPU
DATA AND ADDRESS BUS
HSYNC
VSYNC
PIXEL CLOCK
LINE AND PIXEL COUNTERS
X,Y
ADDRESS AND CONTROL
FIFO
WINDOWS CONTROLLER
CTRL
ADDR
DISPLAY CONTROLLER
ATTR
DECODER
Y, CR, CB
X, Y SYNC
ADDRESS CALCULATOR
ATTRIBUTE MEMORY

FIG. 5

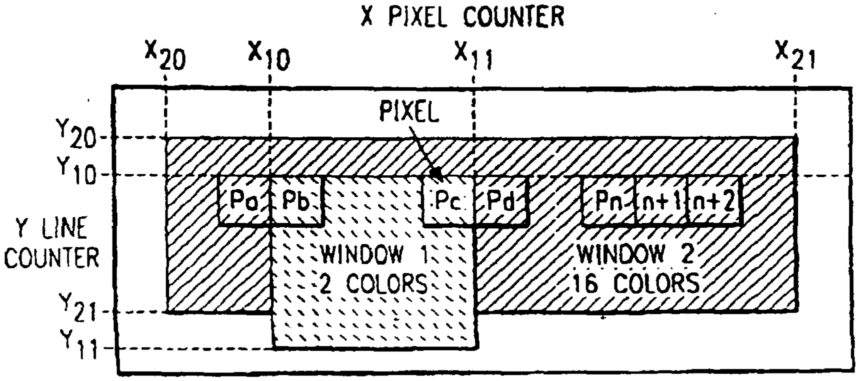

X PIXEL COUNTER
X20
X10
X11
X21
PIXEL
Y20
Y10
Y LINE COUNTER
Y21
Y11
Pa
Pb
Pc
Pd
Pn
n+1
n+2
WINDOW 1
2 COLORS
WINDOW 2
16 COLORS

FIG. 6

FIG. 7

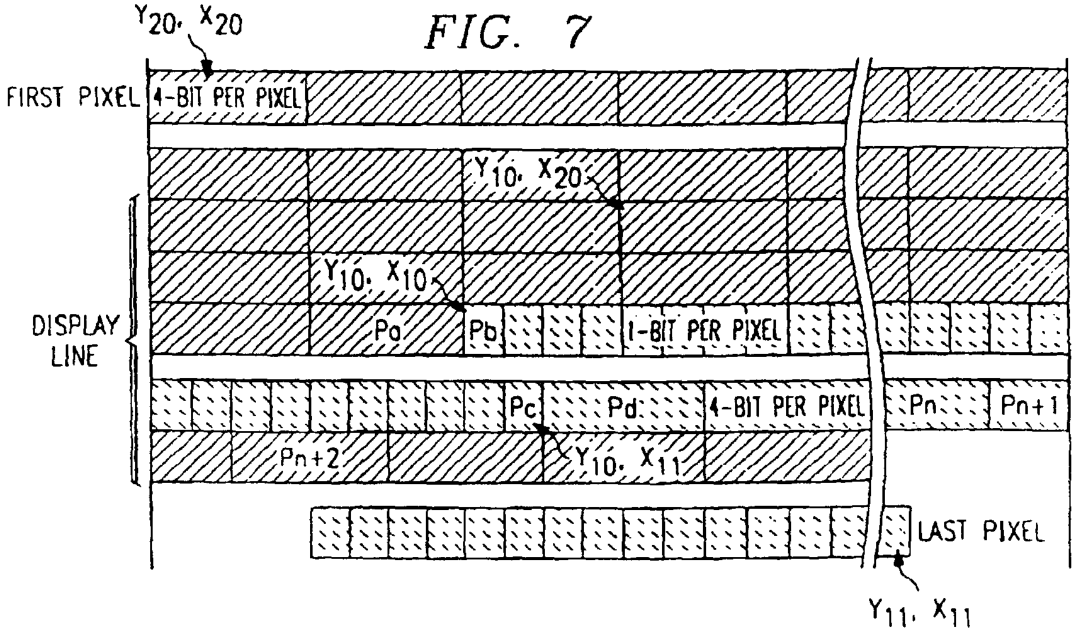
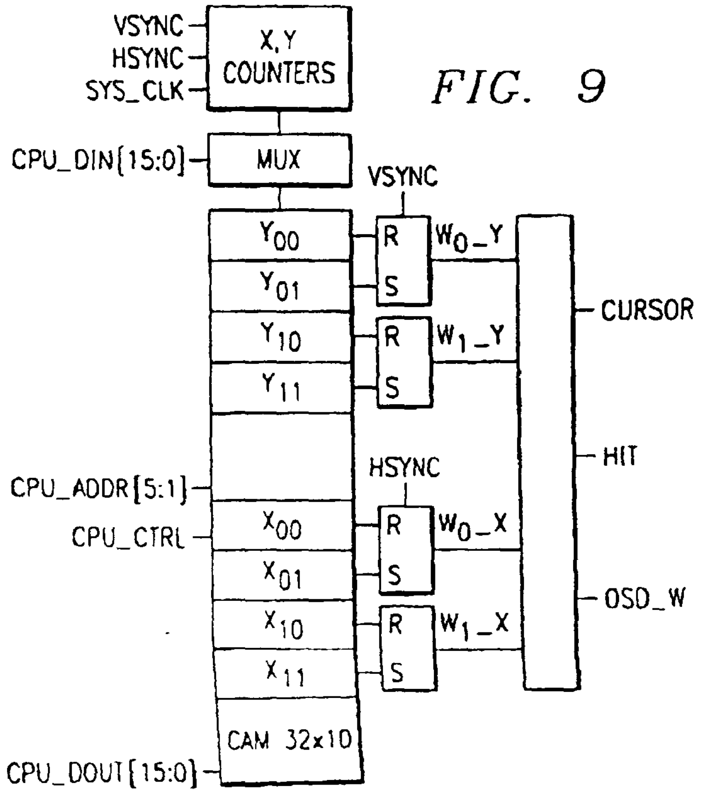

Y_{20}, X_{20}
FIRST PIXEL
4-BIT PER PIXEL
Y_{10}, X_{20}
Y_{10}, X_{10}
DISPLAY LINE
Pa
Pb
1-BIT PER PIXEL
Pc
Pd
4-BIT PER PIXEL
Pn
Pn+1
Pn+2
Y_{10}, X_{11}
LAST PIXEL
Y_{11}, X_{11}

FIG. 9


VSYNC
HSYNC
SYS_CLK
X,Y COUNTERS
CPU_DIN[15:0]
MUX
VSYNC
Y_{00}
Y_{01}
Y_{10}
Y_{11}
R
S
W_0_Y
R
S
W_1_Y
CURSOR
HSYNC
HIT
CPU_ADDR[5:1]
CPU_CTRL
X_{00}
X_{01}
X_{10}
X_{11}
R
S
W_0_X
R
S
W_1_X
OSD_W
CAM 32x10
CPU_DOUT[15:0]

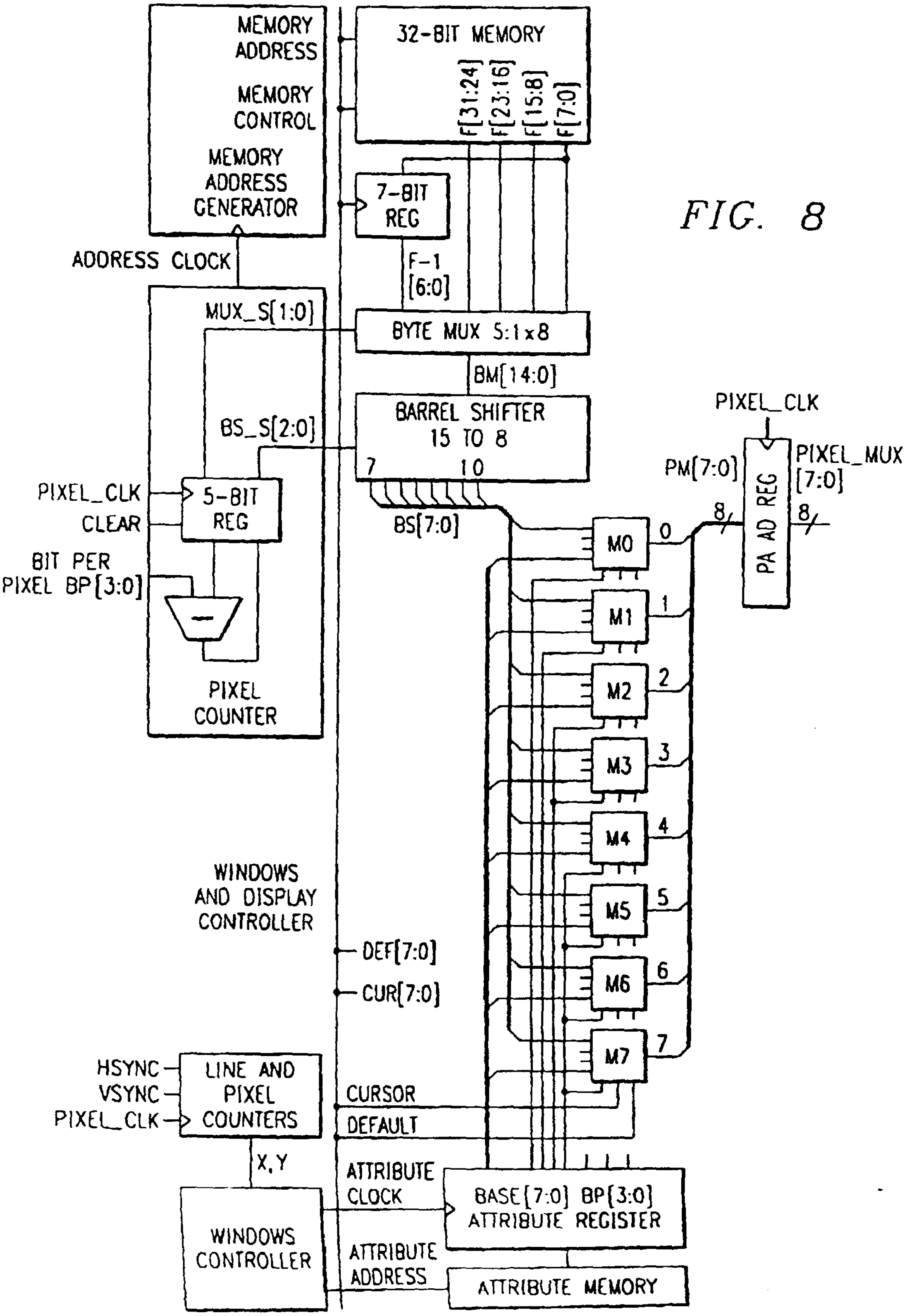

MEMORY ADDRESS
MEMORY CONTROL
MEMORY ADDRESS GENERATOR
32-BIT MEMORY
F[31:24]
F[23:16]
F[15:8]
F[7:0]
7-BIT REG
ADDRESS CLOCK
F-1 [6:0]
MUX_S[1:0]
BYTE MUX 5:1x8
BM[14:0]
BARREL SHIFTER 15 TO 8
7
1 0
BS_S[2:0]
PIXEL_CLK
CLEAR
5-BIT REG
BIT PER PIXEL BP[3:0]
—
PIXEL COUNTER
BS[7:0]
PIXEL_CLK
PM[7:0]
PIXEL_MUX [7:0]
PA AD REG
8
8
M0
M1
M2
M3
M4
M5
M6
M7
0
1
2
3
4
5
6
7
WINDOWS AND DISPLAY CONTROLLER
DEF[7:0]
CUR[7:0]
HSYNC
VSYNC
PIXEL_CLK
LINE AND PIXEL COUNTERS
CURSOR
DEFAULT
X,Y
ATTRIBUTE CLOCK
BASE[7:0] BP[3:0] ATTRIBUTE REGISTER
WINDOWS CONTROLLER
ATTRIBUTE ADDRESS
ATTRIBUTE MEMORY

FIG. 8

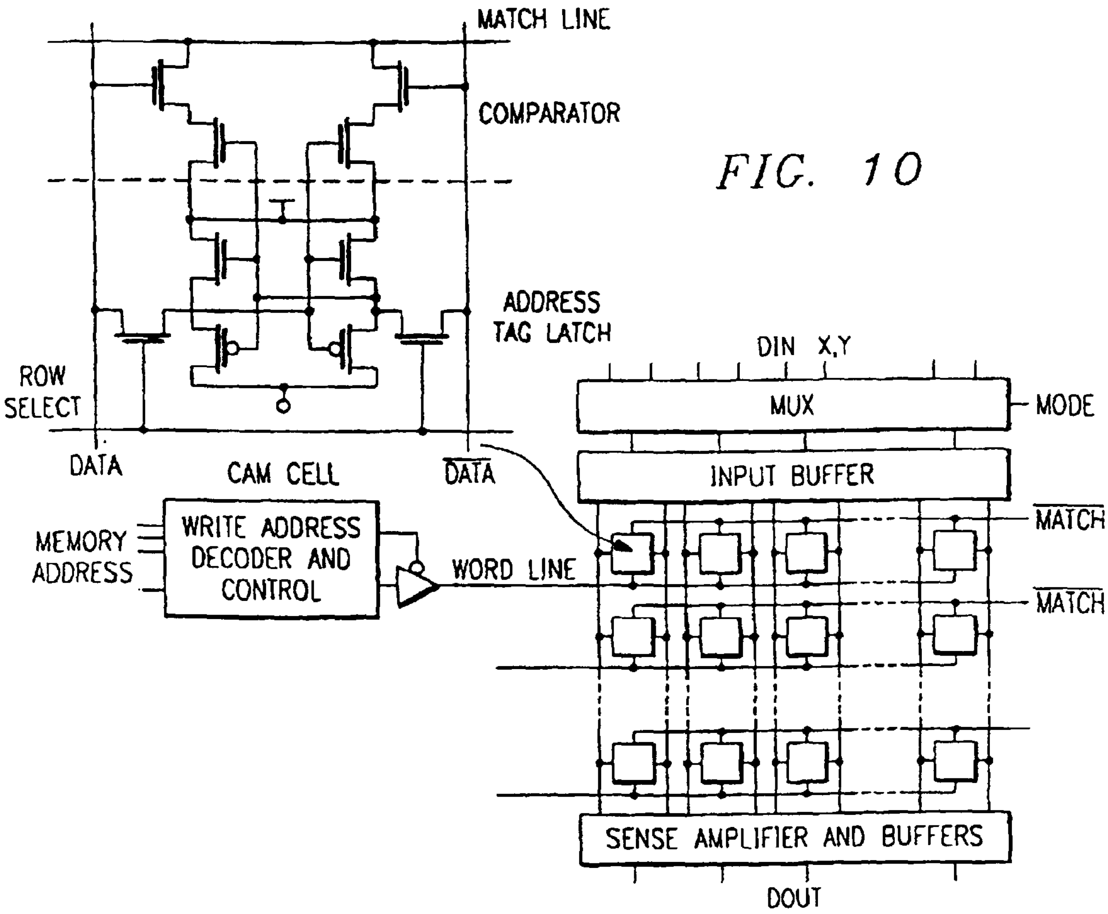

MATCH LINE
COMPARATOR
ADDRESS TAG LATCH
ROW SELECT
DATA
CAM CELL
$\overline{DATA}$
MEMORY ADDRESS
WRITE ADDRESS DECODER AND CONTROL
WORD LINE
DIN X,Y
MUX
MODE
INPUT BUFFER
$\overline{MATCH}$
$\overline{MATCH}$
SENSE AMPLIFIER AND BUFFERS
DOUT

FIG. 10

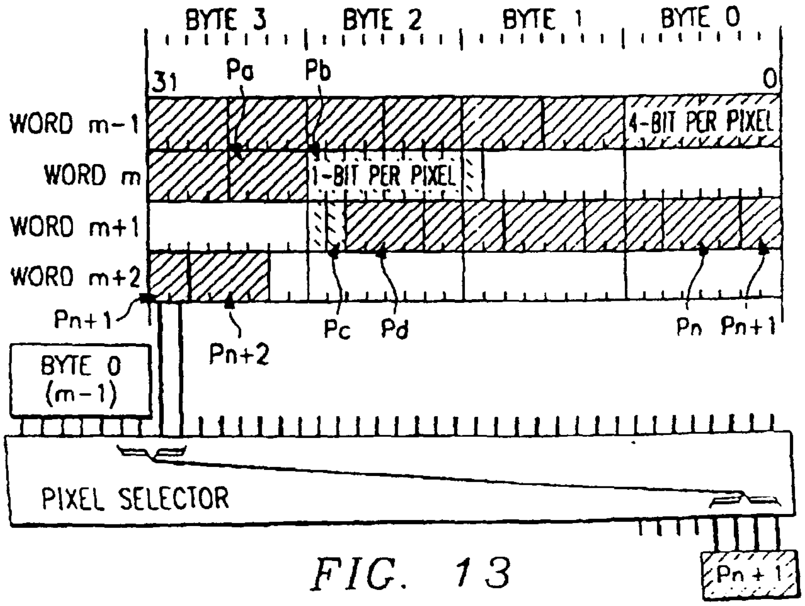

BYTE 3
BYTE 2
BYTE 1
BYTE 0
31
Pa
Pb
0
WORD m-1
WORD m
WORD m+1
WORD m+2
4-BIT PER PIXEL
1-BIT PER PIXEL
Pn+1
Pn+2
Pc
Pd
Pn
Pn+1
BYTE 0 (m-1)
PIXEL SELECTOR
Pn+1

FIG. 13

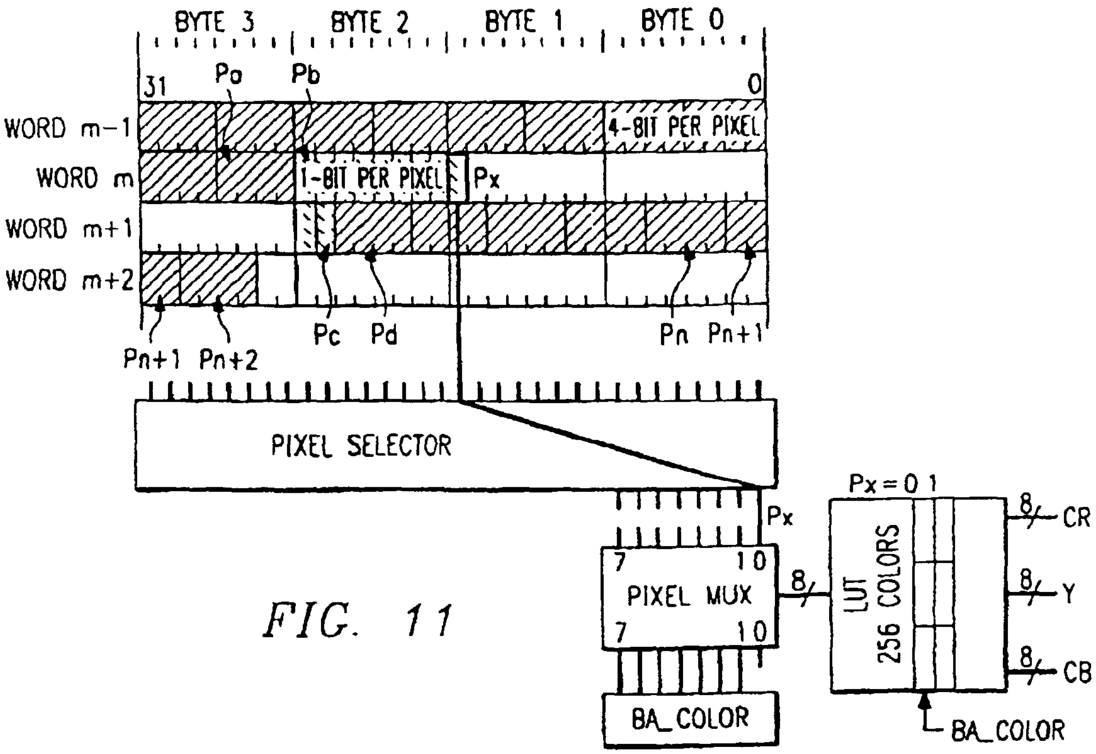

BYTE 3
BYTE 2
BYTE 1
BYTE 0
31
Pa
Pb
0
WORD m-1
WORD m
WORD m+1
WORD m+2
4-BIT PER PIXEL
1-BIT PER PIXEL
Px
Pc
Pd
Pn
Pn+1
Pn+1
Pn+2
PIXEL SELECTOR
Px
Px=0 1
7
1 0
PIXEL MUX
7
1 0
8
LUT
256 COLORS
8
CR
8
Y
8
CB
BA_COLOR
BA_COLOR

FIG. 11

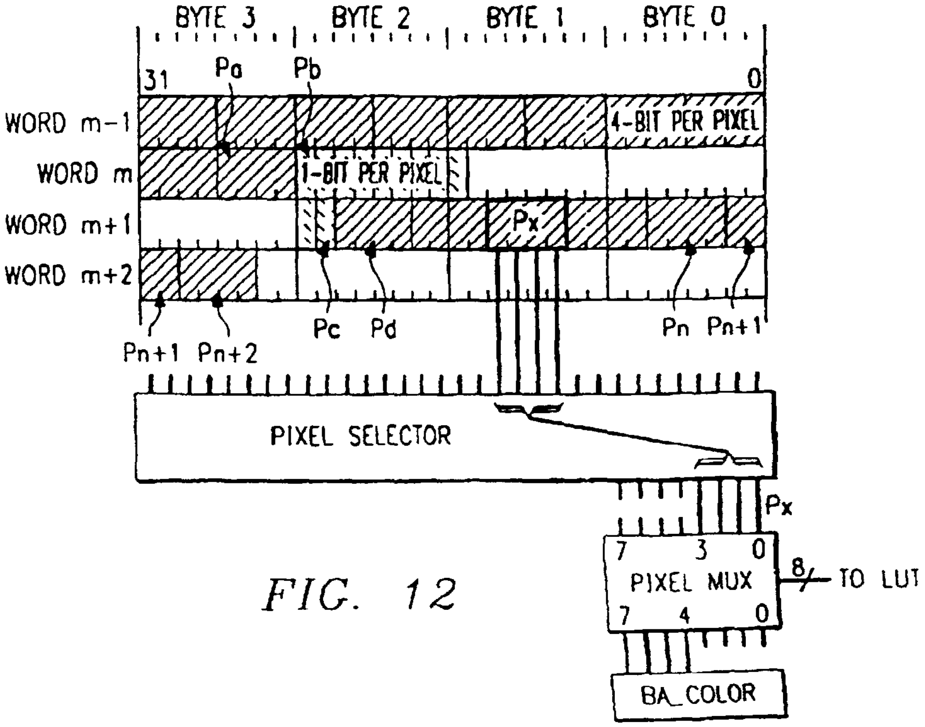

BYTE 3
BYTE 2
BYTE 1
BYTE 0
31
Pa
Pb
0
WORD m-1
WORD m
WORD m+1
WORD m+2
4-BIT PER PIXEL
1-BIT PER PIXEL
Px
Pc
Pd
Pn
Pn+1
Pn+1
Pn+2
PIXEL SELECTOR
Px
7
3
0
PIXEL MUX
7
4
0
8
TO LUT
BA_COLOR

FIG. 12

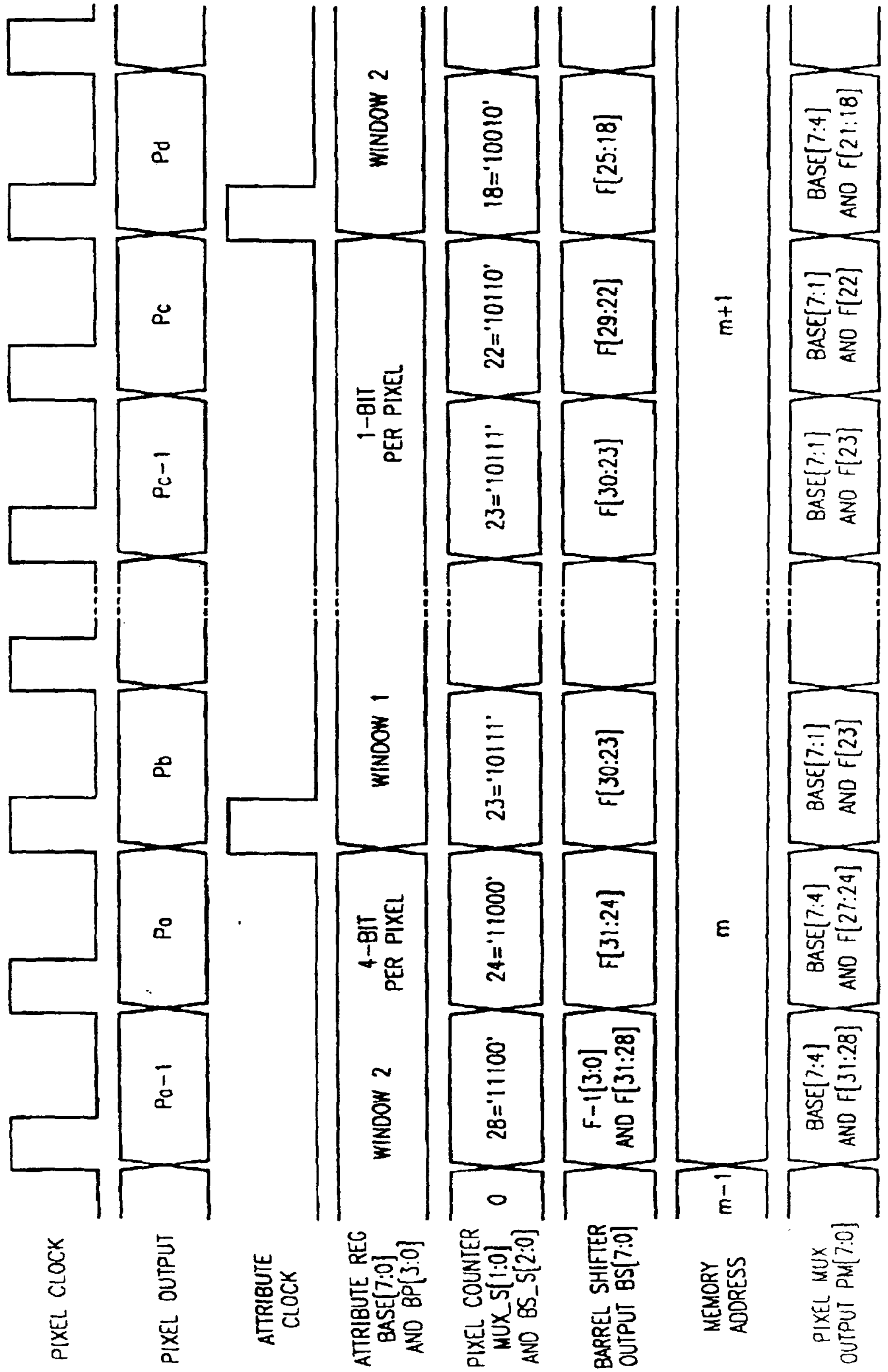

PIXEL CLOCK
PIXEL OUTPUT
ATTRIBUTE CLOCK
ATTRIBUTE REG BASE[7:0] AND BP[3:0]
PIXEL COUNTER MUX_S[1:0] AND BS_S[2:0]
BARREL SHIFTER OUTPUT BS[7:0]
MEMORY ADDRESS
PIXEL MUX OUTPUT PM[7:0]
Pa-1
Pa
Pb
Pc-1
Pc
Pd
WINDOW 2
4-BIT PER PIXEL
WINDOW 1
1-BIT PER PIXEL
WINDOW 2
0
28='11100'
24='11000'
23='10111'
23='10111'
22='10110'
18='10010'
F-1[3:0] AND F[31:28]
F[31:24]
F[30:23]
F[30:23]
F[29:22]
F[25:18]
m-1
m
m+1
BASE[7:4] AND F[31:28]
BASE[7:4] AND F[27:24]
BASE[7:1] AND F[23]
BASE[7:1] AND F[23]
BASE[7:1] AND F[22]
BASE[7:4] AND F[21:18]

FIG. 14a

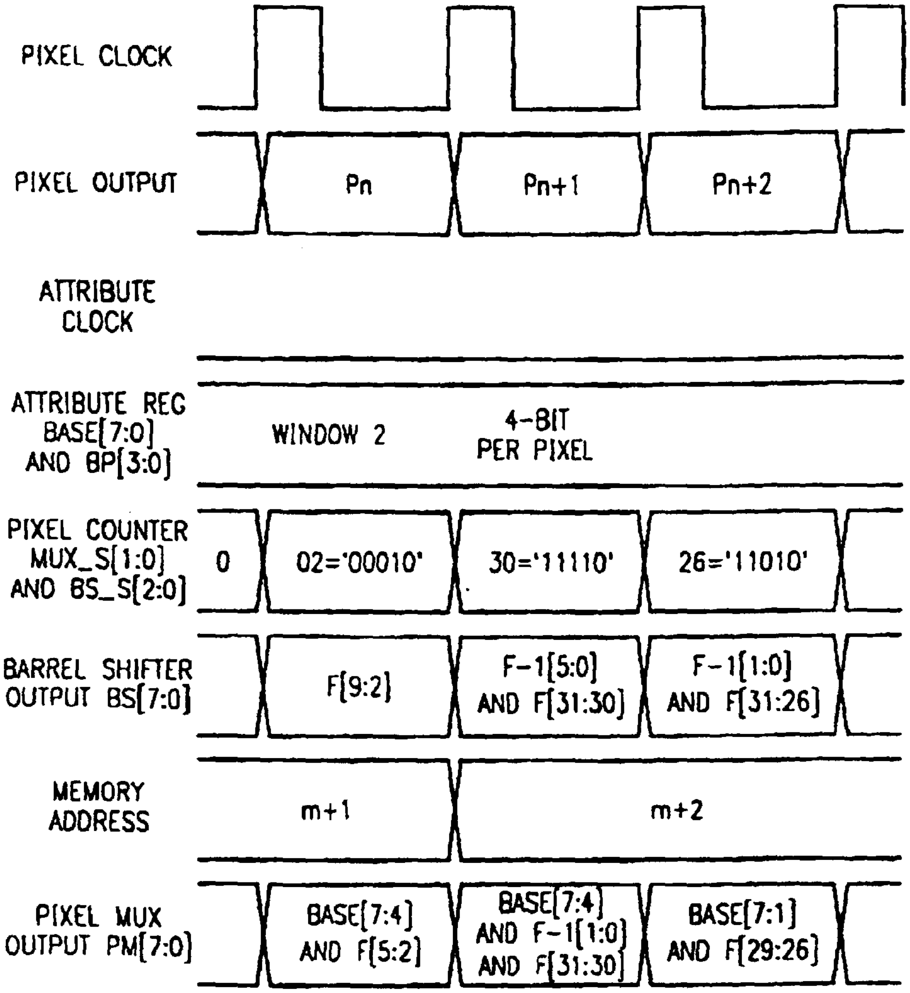

PIXEL CLOCK
PIXEL OUTPUT
Pn
Pn+1
Pn+2
ATTRIBUTE CLOCK
ATTRIBUTE REG BASE[7:0] AND BP[3:0]
WINDOW 2
4-BIT PER PIXEL
PIXEL COUNTER MUX_S[1:0] AND BS_S[2:0]
0
02='00010'
30='11110'
26='11010'
BARREL SHIFTER OUTPUT BS[7:0]
F[9:2]
F-1[5:0] AND F[31:30]
F-1[1:0] AND F[31:26]
MEMORY ADDRESS
m+1
m+2
PIXEL MUX OUTPUT PM[7:0]
BASE[7:4] AND F[5:2]
BASE[7:4] AND F-1[1:0] AND F[31:30]
BASE[7:1] AND F[29:26]

FIG. 14b